UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,058	08/21/2003	J. Patrick Thompson	MSFT-1748/302722.01	1588

41505 7590 06/04/2007

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
PHAM, MICHAEL	

ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
06/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/647,058	THOMPSON ET AL.	
	Examiner	Art Unit	
	Michael D. Pham	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 March 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,13-21,23-25,27-31 and 33-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11,13-21,23-25,27-31 and 33-60* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 3/16/07 has been entered.

Status of claims

2. Claims 1, 21, 37,43, 49, and 55 have been amended.

3. Claims 12, 22, 26, and 32 have been cancelled.

4. Claims 1-11, 13-21, 23-25, 27-31, and 33-60 are pending.

Specification

5. The disclosure is objected to because of the following informalities: please provide application numbers in specifications if already assigned. Example: 10/646941 is the assigned application number to "Systems and methods for separating units of information manageable by a hardware/software interface system from their physical organization.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1 – 3, 5, 7, 8, 10, and 13 – 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) further in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046), **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7,158,962).

8. Regarding claim 1, **Anglin** teaches a computer system comprising: a plurality of Items [storage object], where each of said plurality of Items constitute a discrete storable unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0020] "FIG. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database..." In **Anglin**, the "Item" of the claim is referred to as a 'storage object');

a plurality of Item Folders that constitute an organizational structure for said Items, (See page 2, paragraph [0024] "Storage objects [items] may be defined as part of a storage group [item folder] by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry." In **Anglin**, the "Item Folder" of the claim is referred to as a "storage group"), each of said Items belonging to at least one of the Item Folders (See page 2, paragraph [0024] "Moreover, with the described implementations,

each storage object is capable of being associated with a plurality of different group types."); and

a hardware/software interface system for manipulating said plurality of Items. (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...").

Anglin does not explicitly teach a plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories.

However **Chang** teaches plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..." Also, at least one of the items will belong to the category, by the nature of the objects being the result of the query.) It would have been obvious to one with ordinary skill in the art to combine the computer system as disclosed in **Anglin** with the Categories as disclosed in **Chang** because

of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include a plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories.

Also, **Anglin** and **Chang** do not explicitly teach the Item Folders and the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Item Folders and the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Item Folders and Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the system allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Item Folders and the Categories arranged in a directed graph structure.

Anglin, Chang, and Goodwin do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin, Chang, and Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

9. Regarding claim 2, **Anglin** teaches an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically

result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

10. Regarding claim 3, **Anglin** teaches an item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

11. Regarding claim 5, **Anglin** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

12. Regarding claim 7, **Anglin** teaches each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

13. Regarding claim 8, **Anglin** teaches each said Item is itself automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

14. Regarding claim 10, **Anglin** teaches each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to

delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

15. Regarding claim 13, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach a Category is defined by an Item property. However, **Chang** teaches a Category is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have a Category is defined by an Item property.

16. Regarding claim 14, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category. However, **Chang** teaches one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific

Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.

17. Regarding claim 15, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, **Chang** teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that

are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

18. Regarding claim 16, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. However, **Chang** teaches an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the

disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.

19. Regarding claim 17, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach each of said plurality of Categories is defined by an Item property. However, **Chang** teaches each of said plurality of Categories is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property.

20. Regarding claim 18, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach each of said plurality of Categories is defined by an Item property, and only Items

comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. However, **Chang** teaches each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from amongsaid plurality of Categories can be members of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

21. Regarding claim 19, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories. However, **Chang** teaches each Item comprising the Item property for one of said plurality of Categories are automatically

members of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories.

22. Regarding claim 20, **Anglin** teaches a computer system substantially as shown. **Anglin** fails to teach all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. However, **Chang** teaches all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature

of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.

23. Claims 4, 6, 9 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin,** in view of **Chang, Goodwin**, and **Nelson** as applied to claim 1 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

24. Regarding claim 4, **Anglin, Chang, Goodwin, and Nelson** teach a computer system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** do not explicitly teach an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be

retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

25. Regarding claim 6, **Anglin, Chang, Goodwin, and Nelson** teach a computer system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** do not explicitly teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards**

by simply adding the recycling method of **Edwards** to the system of **Anglin.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

26. Regarding claim 9, **Anglin, Chang, Goodwin, and Nelson** teach a computer system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** do no explicitly teach each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. However, **Edwards** teaches each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been

motivated to have each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.

27. Regarding claim 11, **Anglin, Chang, Goodwin, and Nelson** teach a computer system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** do not explicitly teach each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. However, **Edwards** teaches each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder.

28. Claims 21, 27, 28, 30, and 33 – 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046), **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7,158,962).

29. Regarding claim 21, **Anglin** teaches a hardware/software interface system capable of manipulating an Item of a plurality of Items[data objects] (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art..." Because 'data objects' is plural, it is interpreted that there are a plurality of items.); said Item comprising a discrete unit of information comprising a basic set of properties commonly supported across objects exposed by an operating system shell. (See page 2, paragraph [0020] "FIG. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database..." In **Anglin et al.**, the "Item" of the claim is referred to as a "storage object"), said Item [objects] being a fundamental unit of information manipulated by an operating system (See page 1, paragraph [0005] "In the prior art, the storage management program may define a group as specific objects to be managed, such as data files, database files, programs, etc." This list of objects define fundamental units of information.), a member of one of a plurality of Item

Folders [groups] (See page 2, paragraph [0024] "Moreover, with the described implementations, each storage object is capable of being associated with a plurality of different group types." and see page 2, paragraph [0019] "If the storage objects are associated with an object group, then indication of the associated object group would be included with the object information in the storage database..." In the reference, the "Item" of the instant application is called the "storage object" and the "Item folder" of the instant application is called the "object group" or in some cases the "storage group").

Anglin does not explicitly teach a member of one of a plurality of Categories that constitute an additional organizational structure for said Items.

However **Chang** teaches a member of one of a plurality of Categories that constitute an additional organizational structure for said items. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query...") It would have been obvious to one with ordinary skill in the art to combine the hardware/software interface system as disclosed in **Anglin** with the Categories as disclosed in **Chang** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include a

member of one of a plurality of Categories that constitute an additional organizational structure for said items.

Also, **Anglin** and **Chang** do not explicitly teach the Item Folders and the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Item Folders and the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Item Folders and Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the system allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Item Folders and the Categories arranged in a directed graph structure.

Anglin, **Chang**, and **Goodwin** do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves

Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin**, **Chang**, and **Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

30. Regarding claim 27, **Anglin** teaches said Item is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensures that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any

further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

31. Regarding claim 28, **Anglin** teaches said item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

32. Regarding claim 30, **Anglin** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

33. Regarding claim 33, **Anglin** teaches a hardware/software interface system substantially as shown. **Anglin** fails to teach a Category is defined by an Item property. However, **Chang** teaches said Category is defined by an Item property. (See column 13, lines 26-30 "In the

preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property.

34. Regarding claim 34, **Anglin** teaches a hardware/software interface system substantially as shown. **Anglin** fails to teach said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. However, **Chang** teaches said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to

combine the system as disclosed in **Anglin** with the disclosure of **Chang** because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category.

35. Regarding claim 35, **Anglin** teaches a hardware/software interface system substantially as shown. **Anglin** fails to teach an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, **Chang** teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

36. Regarding claim 36, **Anglin** teaches a hardware/software interface system substantially as shown. **Anglin** fails to teach an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. However, **Chang** teaches an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in **Anglin** with the disclosure of **Chang** because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties.

37. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin**, in view of **Chang, Goodwin, and Nelson** as applied to claim 21 above, and further in view of

Beauregard et al. (hereinafter **Beauregard**, US 6,438,545). **Anglin, Chang, Goodwin, and Nelson** teach a hardware/software interface system substantially as claimed. Anglin et al. fail to teach said Item is a fundamental unit of information manipulated by a virtual machine manager. However, **Beauregard** teaches said Item is a fundamental unit of information manipulated by a virtual machine manager. (See column 13, lines 12-16 "This broad I/O capability can be provided under the Virtual Machine Manager (VMM) that is available under Win32. The VMM is an extensible operating system whose core and standard components are provided by Microsoft Corporation.") Because of the advantages provided by VMM as taught in **Beauregard**, such as the broad I/O capability, it would have been obvious to one with ordinary skill in the art to combine the VMM of **Beauregard** with the teaching of **Anglin, Chang, Goodwin, and Nelson.** It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a virtual machine manager.

38. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin, Chang, Goodwin, and Nelson** as applied to claim 21 above, and further in view of **Gordon and Syme** ("Typing a Multi-Language Intermediate Code" POPL '01 London, UK - Hereinafter **Gordon**). **Anglin, Chang**, and **Goodwin** teach a hardware/software interface system substantially as shown. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item is a fundamental unit of information manipulated by a Common Language Runtime. However, **Gordon** teaches said Item is a fundamental unit of information manipulated by a Common Language Runtime. (See conclusion p. 257 "One of the innovations in Microsoft's Common Language Runtime is

support for typed stack pointers, for passing arguments and results by reference, for example. We presented formal typing rules and a type safety result for a substantial fragment of Common Language Runtime intermediate language. Our treatment of value types and pointer types appears to be new.") It would have been obvious to one with ordinary skill in the art to combine the teaching of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of **Gordon** because of the motivation of being able to pass the arguments and results by reference to have more efficient processing of the data, less transfer overhead, and to be able to handle more robust types of objects. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Common Language Runtime.

39. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin, Chang, Goodwin, and Nelson** as applied to claim 21 above, and further in view of **Judge et al.** (US 6,430,564). **Anglin, Chang, Goodwin, and Nelson** teach a hardware/software interface system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item is a fundamental unit of information manipulated by a virtual machine manager. However, **Judge et al.** teaches said Item is a fundamental unit of information manipulated by a Java Virtual Machine. (See abstract "A data manager manages global data within a Java Virtual Machine (JVM) installed and running in an embedded device. The data manager maintains a data class list that stores data class identifiers associated with each data class object currently loaded and cached in a data cache in the embedded device.") It would have been obvious to one with ordinary skill in the art to combine **Anglin, Chang, Goodwin, and Nelson** with **Judge et al.** by

using the JVM to allow for more diverse types of objects to be processed by the hardware/software interface system. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Java Virtual Machine.

40. Claims 29 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin, Chang, Goodwin, and Nelson** as applied to claim 27 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

41. Regarding claim 29, **Anglin, Chang, Goodwin, and Nelson** teach a hardware/software interface system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the

potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

42. Regarding claim 31, **Anglin, Chang, Goodwin, and Nelson** teach a hardware/software interface system substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin et al.** **Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it

is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

43. Claims 37 – 39 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046), **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7,158,962).

44. Regarding claim 37, **Anglin** teaches a method for organizing Items in computer system, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art..."); said method comprising means by which an Item can be a member of at least two Item Folders (See page 3, paragraph [0026] "Many type of storage object management operations would have to take into account group characteristics and that a storage object may be a member of multiple group types."); but is not owned by any of said Item Folders such that the deletion of any of said Item Folders does not automatically result in the deletion of said Item (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after

eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Anglin does not explicitly teach means by which said Item can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items.

However **Chang** teaches means by which said Item can be a member of a category of a plurality of Categories that constitute an additional organizational structure for said items (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..." Also, at least one of the items will belong to the category, by the nature of the objects being the result of the query.) It would have been obvious to one with ordinary skill in the art to combine the method as disclosed in **Anglin** with the Categories as disclosed in **Chang** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include means by which said Item can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items.

Also, **Anglin** and **Chang** do not explicitly teach the Item Folders and the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Item Folders and the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Item Folders and Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the method allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Item Folders and the Categories arranged in a directed graph structure.

Anglin, **Chang**, and **Goodwin** do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the

recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin, Chang**, and **Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

45. Regarding claim 38, **Anglin** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

46. Regarding claim 39, **Anglin** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

47. Regarding claim 41, **Anglin** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

48. Claims 40 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin,** in view of **Chang, Goodwin, and Nelson** as applied to claim 38 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

49. Regarding claim 40, **Anglin, Chang, Goodwin, and Nelson** teach a method substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it no longer

belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

50. Regarding claim 42, **Anglin, Chang, Goodwin, and Nelson** teach a method substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than

delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

51. Claims 43 – 45, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046), **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7,158,962).

52. Regarding claim 43, **Anglin** teaches a computer-readable medium comprising computer-readable instructions for an Item of a plurality of Items, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client

programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art...")

a plurality of Item Folders that constitute an organizational structure for said Items, (See page 2, paragraph [0024] "Storage objects [items] may be defined as part of a storage group [item folder] by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry." In **Anglin**, the "Item Folder" of the claim is referred to as a "storage group"), each of said Items belonging to at least one of the Item Folders (See page 2, paragraph [0024] "Moreover, with the described implementations, each storage object is capable of being associated with a plurality of different group types.")

Anglin does not explicitly teach said Item being a member of one of a plurality of Categories that constitute an organizational structure for said Items.

However **Chang** teaches said Item being a member of one of a plurality of Categories that constitute an organizational structure for said Items (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..." Also, at least one of the items will belong to the category, by the nature of the objects being the result of the query.) It would have been obvious to one with ordinary skill in the art to combine the method as disclosed in **Anglin** with the

Categories as disclosed in **Chang** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include said Item being a member of one of a plurality of Categories that constitute an organizational structure for said Items

Also, **Anglin** and **Chang** do not explicitly teach the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the method allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Categories arranged in a directed graph structure.

Anglin, **Chang**, and **Goodwin** do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin**, **Chang**, and **Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

53. Regarding claim 44, **Anglin** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

54. Regarding claim 45, **Anglin** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

55. Regarding claim 47, **Anglin** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the

"group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

56. Claims 46 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin,** in view of **Chang, Goodwin, and Nelson** as applied to claim 44 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

57. Regarding claim 46, **Anglin, Chang, Goodwin, and Nelson** teach a computer readable medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said

Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

58. Regarding claim 48, **Anglin, Chang, Goodwin, and Nelson** teach a computer readable-medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

59. Claims 49 – 51 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046) and in view of **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7,158,962).

60. Regarding claim 49, **Anglin** teaches a computer readable medium comprising computer-readable instructions for organizing Items in computer system, said Items comprising a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art..."); said method comprising means by which an Item can be a member of at least two Item Folders (See page 3, paragraph [0026] "Many type of storage object management operations would have to take into account group characteristics and that a storage object may be a member of multiple group types."); but is not owned by any of said Item Folders such that the deletion of any of said Item Folders does not automatically result in the deletion of said Item (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Anglin does not explicitly teach means by which said Item can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items.

However **Chang** teaches means by which said Item can be a member of a category of a plurality of Categories that constitute an additional organizational structure for said items (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..." Also, at least one of the items will belong to the category, by the nature of the objects being the result of the query.) It would have been obvious to one with ordinary skill in the art to combine the method as disclosed in **Anglin** with the Categories as disclosed in **Chang** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include means by which said Item can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items.

Also, **Anglin** and **Chang** do not explicitly teach the Item Folders and the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Item Folders and the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-

many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Item Folders and Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the method allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Item Folders and the Categories arranged in a directed graph structure.

Anglin, **Chang**, and **Goodwin** do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin, Chang,** and **Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

61. Regarding claim 50, **Anglin** teaches the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

62. Regarding claim 51, **Anglin** teaches the item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

63. Regarding claim 53, **Anglin** teaches the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

64. Claims 52 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin,** in view of **Chang, Goodwin, and Nelson** as applied to claim 50 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

65. Regarding claim 52, **Anglin, Chang, Goodwin, and Nelson** teach a computer readable medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item

Folder. However, **Edwards** teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

66. Regarding claim 54, **Anglin, Chang, Goodwin, and Nelson** teach a computer-readable medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or

delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

67. Claims 55 – 57 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin et al.** (hereinafter **Anglin**, US 2004/0199521) in view of **Chang et al.** (hereinafter **Chang**, US 6,578,046), **Goodwin et al.** (hereinafter **Goodwin**, US 6,199,195), and **Nelson** (hereinafter **Nelson**, US 7158962).

68. Regarding claim 55, **Anglin** teaches a computer-readable medium comprising computer-readable instructions for a hardware/software interface system, said operating system comprising: means for manipulating a plurality of Items comprising at least one Item, where each of said plurality of Items constitute a discrete unit of information that can be manipulated by a hardware/software interface system (See page 2, paragraph [0020] "FIG. 2a illustrates the

data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database..." In **Anglin**, the "Item" of the claim is referred to as a "storage object"); means for manipulating a plurality of Item Folders comprising at least one Item Folder, wherein said plurality of Item Folders constitute an organizational structure for said Items (See page 2, paragraph [0024] "Storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry." In **Anglin**, the "Item Folder" of the claim is referred to as a "storage group"); and wherein each of said plurality of Items belongs to at least one of said plurality of Item Folders, and wherein each of said plurality of Items may belong to more than one Item Folder of said plurality of Item Folders (See page 2, paragraph [0018] "The Server includes a storage management server program that is capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, archival operations, hierarchical storage management related operations or any type of storage management operations known in the art..." and see page 2, paragraph [0019] "If the storage objects are associated with an object group, then indication of the associated object group would be included with the object information in the storage database..." and see page 2, paragraph [0024] "...by adding the group identifier of the one or more groups in which the storage object is a member..." – In other words, the "Item" ["storage object"] may belong to more than one folder ["storage group"]).

Anglin does not explicitly teach means for manipulating a plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories.

However **Chang** teaches plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." This is how Categories are defined in the instant specification in paragraph [0110] p. 38 "Conceptually, Categories can also be thought of as virtual Item Folders whose membership is based on the results of a specific query..., and Items that meet the conditions of this query..." Also, at least one of the items will belong to the category, by the nature of the objects being the result of the query.) It would have been obvious to one with ordinary skill in the art to combine the computer system as disclosed in **Anglin** with the Categories as disclosed in **Chang** because of the advantage of having the relationships stored as part of the object rather than having to execute a query to form the relationships each time. It is for this reason that one of ordinary skill in the art would have been motivated to include a plurality of Categories that constitute an additional organizational structure for said items, at least one of said Items belonging to at least one of the Categories.

Also, **Anglin** and **Chang** do not explicitly teach the Item Folders and the Categories arranged in a directed graph structure.

However **Goodwin** teaches the Item Folders and the Categories arranged in a directed graph structure. (See column 4, lines 30 – 31 and lines 34 – 44 "A 'relationship' defines a link between two object classes...Relationships can be one-to-one, one-to-many, or many-to-many....". A directed graph structure is represented by relationships as described in Goodwin, therefore without mentioning the words directed or graph, they are referring to the same thing. Also Item Folders and Categories are essentially defined in the specification to be version of object classes.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Anglin** and **Chang** with that of **Goodwin** because they are all related to managing objects, and by including the directed graph structure as disclosed in Goodwin, the system allows for more than hierarchical combinations and for more divers relationships between the objects. It is for this reason that one of ordinary skill in the art would have been motivated to include the Item Folders and the Categories arranged in a directed graph structure.

Anglin, **Chang**, and **Goodwin** do not explicitly teach the ",wherein said Item Folder are themselves Items and said Items can be modified via any Item Folder they belong to;"

However, **Nelson** discloses col. 3 lines 59-60 that an Item (item) can be for example a folder (item folder) or document. Hence suggesting "wherein said Item folder are themselves Items". Further disclosing in the abstract that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." That is, moves an item from one folder to another suggests "said items can be modified via any item folder they belong to". To clarify, **Nelson** suggests the

recited "said items" (items) "can be modified via any Item Folder they belong to" (moves an item from one folder to another).

Accordingly, **Anglin, Chang, Goodwin, and Nelson** are all related to managing objects, and are therefore all within the same field of endeavor. For the above reasons, it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply **Nelson's** disclosure of col. 3 lines 59-60 that an "Item can be for example a folder or document"; and from abstract, that the system "automatically moves an item from one folder to another when the item is changed to a different type if the item was first added to the folder by the present system." to the disclosure **Anglin, Chang,** and **Goodwin** in order to in order to enhance the systems by supporting multiple levels of folders and multiple attributes. Hence, automating the task of linking items to folders and improving performance for multiple level nested folders.

69. Regarding claim 56, **Anglin** teaches an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which

represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

70. Regarding claim 57, **Anglin** teaches an item is automatically deleted when it no longer belongs to any Item Folder. (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

71. Regarding claim 59, **Anglin** teaches said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted. (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

72. Claims 58 and 60 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Anglin,** in view of **Chang, Goodwin, and Nelson** as applied to claim 56 above, and further in view of **Edwards** (US Patent Application Publication 2004/0073560).

73. Regarding claim 58, **Anglin, Chang, Goodwin, and Nelson** teach a computer-readable medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, **Edwards** teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

74. Regarding claim 60, **Anglin, Chang, Goodwin and Nelson** teach a computer-readable medium substantially as claimed. **Anglin, Chang, Goodwin, and Nelson** fail to teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, **Edwards** teaches said Item, when it is a

member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of **Anglin, Chang, Goodwin, and Nelson** with the disclosure of a recycling method of **Edwards** by simply adding the recycling method of **Edwards** to the system of **Anglin. Edwards** points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Response to Arguments

75. Applicant's arguments with respect to claims 1-11, 13-21, 23-25, 27-31, and 33-60 have been considered but are moot in view of the new ground(s) of rejection.

Applicant's assert the following directed towards the Anglin reference (lettered).

A. That Anglin does not disclose a plurality of "item folders, wherein said item folders are themselves items and said items can be modified via any item folder they belong to." That therefore, the combination does not disclose "a plurality of Item Folders that constitute

organizational structure for said items, each of said Items belonging to at least one of the Item folders, wherein the item folders are themselves items".

In response, the assertions are moot upon grounds of new rejection. The new combination of Anglin, Chang, Goodwin, and Nelson discloses the asserted limitations of claims 1, 21, 37, 43, 49, and 55.

Conclusion

76. The prior art made of record listed on PTO-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

Contact Information

77. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael D. Pham whose telephone number is (571)272-3924. The examiner can normally be reached on Monday - Friday 9am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Pham
Art Unit 2167
Examiner M.P.

Cam Y. Truong
Art Unit 2162
Primary Examiner

CY

John Cottingham
Art Unit 2167
Supervisor

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100